JC08 Rec'd PCT/PTO 15 MAR 2001

LINE OBJECT VECTORIZATION IN COLOUR/GRAYSCALE IMAGES

FIELD OF THE INVENTION

This invention relates to a method and apparatus for the detection, tracing and vectorization of line objects in colour and/or grayscale images. The invention is particularly suitable for the detection of line images in colour/grayscale maps, aerial photographs, satellite images, and line drawings such as engineering and architectural drawings.

BACKGROUND OF THE INVENTION

Image data - including images including line objects - is in many cases captured in a raster data format. Examples of this include scanned paper maps, satellite images, aerial photographs, and scanned line drawings. However if such images include important line objects it is more convenient to have the images converted into a vector format such that processing tools such as Geographic Information Systems (GIS) and Computer Aided Design (CAD) packages can be used to manipulate the data in applications such as land and urban planning, traffic planning and control, building and estate design and management. This conversion of line objects such as roads and contour lines from raster format to vector format is known as vectorization or digitization.

In the digitization of paper maps, for example, the detection and digitization of line objects such as roads, rivers, contour lines and region boundaries are the most time consuming tasks in the digitization process.

Conventional techniques firstly require colour segmentation, assuming that a given class of line objects are all of the same colour. In a map for example, all roads may be brown, or all rivers blue, railway lines black, and so on. By means of colour segmentation a colour map image is converted into several binary images each representing a layer of the map. Interactive line tracing is then performed on the binary images. A user may typically click on one line point and from that point the system will perform a line tracing. The trace will generally stop whenever there is a break or any other problem and the user must click again to re-start the tracing process. For a number of reasons the colour segmentation process usually creates too many line breaks and therefore the tracing process requires substantial user intervention and the vectorization process is very labour-intensive.

Other examples of the prior art can be found in US 5,631,982, US 5,691,827 and US 5,345,547. US 5,631,982 discloses a system in which lines in images are detected using line neighbourhoods and a parallel co-ordinate transformation. Line neighbourhoods are said to accommodate the uncertainty in line detection arising from image noise. In US 5,691,827

the system considers the width of the line and rejects lines having a width equal to or less than a predetermined width. In US 5,345,547 contour line characteristic points are detected using direction information.

SUMMARY OF THE INVENTION

According to the present invention there is provided a method for the vectorization of line objects in a colour or grayscale image comprising the steps of:

(a) collecting sample data of line points on line objects within said image,

(b) extracting multiple features from the collected sample data to represent characteristics of the line points,

(c) grouping said data into a plurality of clusters in a multi-dimensional feature space, each said cluster comprising a plurality of line points having feature measures within a selected criteria set,

(d) detecting further line points by matching image points to said clusters and rejecting image points not falling within any cluster,

(e) performing a line tracing operation based on the detected line points and features; and

(f) identifying and correcting possible errors.

A feature of the described embodiment of the present invention is the collection of sample data which is known to correspond to line points and which may then be used as a

prototype to detect further line points. This sample data is preferably collected interactively by means of a user identifying two points on a line. The line points between the two identified points are automatically identified by the system and selected as samples. The feature measures extracted from collected sample line points may represent a number of features of the line points, for example colour, line profile, line width, line direction and spatial location of the points.

Once obtained the sample data may then be clustered into well-defined clusters in a multi-dimensional feature space. This reduces the possibility of including background image points in the line clusters. The clusters are preferably defined in such a way that the clusters occupy a minimum region in the feature space. By providing such clusters further line points may be detected from the set of image points by comparing the image points with the cluster criteria. If there is a match, an image point is assigned as a line point within that cluster. If an image point is not found to match any cluster, it is rejected as not being a line point. To facilitate the processing time there is preferably a decision process in which the image points are compared with the cluster criteria. For example, the image point may be judged on colour firstly for a match, and if a colour match is found the other features may then be used for verification of the match.

It is important that a good and accurate match is made since the detected line points may then act as seeds for a line tracing algorithm. This line tracing algorithm may be completely automatic or it may be partly interactive. In its automatic mode the algorithm comprises comparing each potential line point within a look-ahead window with a known line point at the end of a line segment and adding the best match line point to the segment. In addition, all possible line points between the best match line point and the end of line segment may also be added to fill the gap between them. If the best match point is itself a point at the end of a line segment, the two line segments may be merged. As an alternative to this process being fully automatic, it may be partly interactive. In this arrangement a user may select a point for line tracing to begin from which tracing will then continue until no further suitable point is found at which point the trace stops until recommenced by the user.

Although the described embodiment of the present invention is capable of providing a robust and highly accurate system, it is nonetheless inevitable that errors will always occur and, accordingly, the described embodiment includes an error identification and correction process. This process is preferably an interactive process in which possible errors are presented to a user who is also provided with a range of operations for correcting and editing the image. These operations include means for smoothing a line, means for

filtering out unwanted image points, and means for recognising and deleting characters, for example map annotations, that may have been included in error. Smoothing for example may involve fitting the line to a curve and redistributing the points along the line more evenly. Another form of correction may be to join together two broken line segments by means of a curve being fitted between them. This curve may preferably be fitted to a number of points along each line segment for the best fit.

The invention also extends to apparatus for vectorization of line objects in a colour or grayscale image comprising, means for semi-automatically collecting sample data of line points on line objects within said image, means for extraction of multi-dimensional feature measures from said sample line points, classifying means for grouping said data into clusters each said cluster having a plurality of line points having feature measures within a selected criteria set, means for comparing image points with said clusters to find image points that match with said clusters and for rejecting image points that do not match with any cluster, means for performing a line tracing operation based on detected line points and features, and means for identifying and correcting errors.

Viewed from another broad aspect the present invention provides a method for the vectorization of line objects in a colour or grayscale image in which sample line points are

used to generate a plurality of prototypes, each said prototype comprising a cluster of line points having parameters within defined ranges, and in which line points are detected from the image by matching image points with said prototypes and assigning an image point to a line point where there is a match and rejecting an image point where there is no match.

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments of the invention will now be described by way of example and with reference to the accompanying drawings, in which:-

Fig. 1 is a block diagram showing the training and detection components of an embodiment of the present invention.

Figs. 2(a)-(c) illustrate the detection of a line profile and centre.

Fig. 3 illustrates the concept of using clusters for feature space optimization.

Fig. 4 is a block diagram illustrating a four-round line tracing algorithm.

Sub A1

Fig. 5 is a flow-chart illustrating a method for finding the next line point.

Fig. 6 is a block diagram illustrating a method for interactive break points linking and editing.

Fig. 7 is a schematic diagram of a line, illustrating

line profile.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

As will be seen in more detail from the following description, in its preferred forms the present invention provides a method for the vectorization of line objects directly from colour or grayscale images without the need for a colour segmentation process. To begin with, sample line points are collected on an interactive basis with the user. The sample data is subjected to optimisation in feature space by being grouped into clusters corresponding to particular criteria and in such a way that the clusters occupy a minimum area in feature space. Further line points are detected by matching against these prototypes and adaptive line tracing commences providing that at least one line point is known. Finally an interactive editing and verification process is performed. In an alternate embodiment, as will be seen further below, the line tracing may be performed on an interactive basis.

When analysing a line in a colour/grayscale image, the human eye considers a number of features of the line. These include the colour of the line, its shape, its width and its direction. In preferred forms of the present invention this is mimicked by the system of the invention which uses multiple features of lines in the process of detection and tracing of line objects. These features include, but are not

limited to (1) the colours of lines, (2) their profiles, (3) the direction of a line segment, and (4) the line width.

(1) The colour of lines may be represented as points in colour space. There are several colour spaces that can be used (see Digital Image Processing, William K. Pratt; Wiley New York, 1978). For example, (L*,a*,b*) colour space is uniform with respect to human perception, while (H,S,V) [Hue, Saturation, Value] colour space is extensively used for graphics. In practice a line will not be of uniform colour. There are normally variations of colour both along and across the width of a line. The colour of a line at a particular point k along a line is computed as an average over a small window centred on k and having the same width as the line. Another dimension of the window, which is along the orientation of the line segment may be 1, 2 or 3 pixels for example.

(2) The profile of a line can be defined as an array of pixel values (colour or gray-level) across the line in a direction perpendicular to the line orientation. The array will be a few pixels wider than the line width, and each element of a profile can be an average of a few adjacent pixels along orientation of the line segment. The profile will form a colour ridge, the peak of the ridge may occur at the line centre. This is illustrated in Fig. 7 showing a position of a line L of width W in which there are seven values P_1 - P_7 in the array, of which P_1, P_2 and P_6 and P_7 are values of background points and P_3 - P_5 are values of line

points, with P_4 being the value of the line centre point.

(3&4) Line width is self-defining, while the direction of a line can be defined as the straight-line which best fits the line segment.

The described embodiment of the present invention allows these features of a line to be used together in line detection and tracing. In particular, a similarity function is used based on a combination of measurements of multiple features to provide a usable measure of likelihood that two points relate to the same line object.

A generic similarity function can be of the form:

$$\mathrm{Sim}(lp_1,lp_2) = g[\mathrm{sim}_{profile}(lp_1,lp_2), \mathrm{sim}_{color}(lp_1,lp_2), \mathrm{sim}_{width}(lp_1,lp_2), \mathrm{sim}_{orientation}(lp_1,lp_2)] \quad \text{....Eq. 1}$$

There g is a generic function which is used to calculate an overall similarity measures between the line point 1 and line point 2 by fusing similarity measures on individual line features of profile, colour, line width and orientation and lp stands for the line point A preferred, specific example of such a function is as follows:

$$\mathrm{Sim}(lp_1,lp_2) = \mathrm{distance}_{colour}(lp_1,lp_2)/\mathrm{correlation}^3_{profile}(lp_1,lp_2) \quad \text{....Eq. 2}$$

...where the colour similarity is calculated by distance measure of two colours in a colour space. It is commonly recognised that in (L, a, b) colour space, the Euclidean distance between two colours coincides well with the perceptual differences. On other hand, normalised correlation of two profiles reflects the difference of the shape of two profiles. Therefore, correlation is used to measure the similarity of profiles of two lines points.

To get the overall similarity between these two line points, here in this equation, the distance is divided by third order of correlation. In the case of two very similar line points, the colour distance will be small, and the correlation will close to 1, so the overall similarity is small. On other hand, for a pair of very distinct points, the colour distance will be large, and the correlation will be much smaller than 1, the cubic and division operands will make the overall similarity measure even large.

For a general discussion of similarity functions, see pp 313 - 316 Neural Networks and Simulations Networks, J. K. Wu, Marcel Dekker, Inc. New York 1994.

One or more of the features (for example colour or width) can be compared using one similarity function to generate hypotheses which can then be verified by others (for example profile and direction) using another similarity function.

Alternatively different features or sets of features can be used in different situations. For example, in the line point recognition process, colour and profile are necessary to make sure the false acceptance rate of the recognition is kept low. In the case of line tracing, because it is necessary to check if the next point is a line point and the direction is a high priority, verification of colour similarity may be enough in many cases.

With reference to Figure 1 a schematic diagram of the method of the described embodiment of the invention is shown. The method comprises a training phase 100 and a detection phase 200.

Training Phase: Sample Data Collection

The first stage of the training phase 100 is that of sample data collection at step 110. A small number of sample line points are collected interactively by the user. Those points should be distributed over different background ares. The user is required to define the line width (w) at the beginning of this operation.

Since it is tedious and time consuming for the user to select the sample line points one by one and the sample data collection method requires the user to identify two line points only by clicking on the line at each location on a line, and the in-between line points will be automatically

located. Since it is very difficult for human users to click exactly on the line center, the method not only automatically finds the line centre for line points in-between two user defined points, but also verifies the centre of these two user identified points as well. This saves a lot of sample data collection time while ensuring the accurate location of the centre of sample line points collected.

To begin with, all possible in-between points along the line AB are determined approximately and then the line centres are located precisely for all candidate line points including the points input by the user. Here, a colour profile ridge as described with reference to Figure 7 is used as the main feature to locate automatically the line centre point. If the line width is one pixel, normally the line centre is at the ridge of the line profile. On other hand, if the line width is more than one pixel, there may not be any ridge at all on the profile or the ridge may not represent the line centre due to noise. In this case, the line centre can be robustly detected at the ridge of the profile's line width average function. This function is calculated by convoluting the profile with a window function, the width of which is equal to the line width W.

Consider, for example C as a colour point along line segment AB as shown in Fig. 2a. The colour profile is calculated in a colour space such as {H,S,V} as follows:

$\{P_{KH}, P_{KS}, P_{KV}\}$, $k = 1, 2, . . . , n$

where n is the dimension of the profile array, and pk is the k'th element of the profile, n being much greater than the line width. The three components of the colour profile, $\{P_{KH}, P_{KS}, P_{KV}\}$ will vary to different extents and for higher accuracy the component which has the largest variations should be chosen. This can be done by calculating the standard deviation for all three components from the first group of samples and the component with maximum standard deviation is used for subsequent sample collection. Assume that P_{KX} is chosen for its maximum deviation, Fig. 2(b) shows the profile at C of component pk^x. This profile may then be convoluted with a window function of a width equal to the line width w and is shown in Fig. 2(c). Fig. 2(c) clearly shows a ridge corresponding to the line centre.

<u>Training Phase: Feature Space Optimization and Prototype Generation</u>

Depending on the size of the image and the number of line objects within it, the user will in the sample data collection stage select an appropriate number of sample points, for example 200, spread over the whole image. These sample points are a representative sample of all possible line points in the image to be traced. From these sample points, multiple features including colours and profiles are extracted. Subsequent line point detection is based on

measures of these features which characterise a number of properties of the line points: eg colour, profile, line width and orientation. Since multiple features of the line points are used as part of the line point detection process, line point detection can be performed with a greater degree of accuracy than in the prior art.

The sample points are subject at step 120 to feature space optimisation. Due to variations in lines across the image, the sample points for the lines may not necessarily appear as a single cluster in the feature space. Therefore a technique of feature space optimization is employed to find a small number of clusters which can optimally represent the characteristics of all possible line points.

For the classical recognition by classification problems such as numerical character recognition, the number of classes and the samples of all numerical characters are available. Here, since the problem is of line object vectorisation, knowledge of the line objects is only available by selecting samples, and it is very difficult to select enough samples which can represent all variations of the background since this varies from location to location in the image and it is not possible to know what the variations are.

Instead of using conventional recognition by classification therefore, so-called recognition by recall is used, in which the sample data is clustered such that a number of clusters

are found including the sample data with the minimum region in feature space at step 130. Suitable algorithms for optimally clustering such data are known in the art. In this embodiment, a hierarchical clustering algorithm of the same type as Centroid (see Multidimensional Clustering Algorithms, by Fionn Murtagh; Physica-Verlag, Vienna 1995) is preferred. The clustering algorithm works by iteratively merging smaller clusters into bigger ones. It starts with one data point per cluster. Then it looks for the smallest distance between any two clusters and merges them into one cluster. Euclidean distance may be used to measure this distance. Preferably, however, similarity functions as discussed above are used. Distance is then evaluated as a similarity measure. The merging is repeated until a termination criterion is reached. Here the termination criterion is defined as to minimise the inner cluster distance and maximise inter-cluster distance (as discussed in the book noted above). After clustering, the clusters with certain population are used as prototypes, and small clusters are removed to reduce the false positives. The mean (M) and deviation (D) are used to represent each cluster. Such clustering may be performed on any feature, but the inventions have found that acceptable results can be obtained by limiting clustering to colour and profile using similarity functions as discussed above to assign points between clusters.

Fig. 3 illustrates by way of example sample points being represented by two clusters in a two-dimensional feature

space although it is to be appreciated that the feature space employed in a practical example would likely be more multi-dimensional.

Thus by this stage in the vectorization process sample data has been collected by means of the user identifying pairs of points on selected lines and by means of the system subsequently calculating the line points between the pairs identified by a user. This data is then organised into clusters in feature space each cluster corresponding to sample points having particular criteria, eg. colour, profile, line width, with the clusters being chosen so as to minimise the space required and these clusters define prototypes for subsequent data as now described.

Detection Phase - Line Detection

sub B1

The first step in the deletion phase is that all the data comprising the image to be vectorized is read at step 210 and each point is matched against the prototypes at step 220. This matching requires a decision-making process to reduce the total detection time. To begin with, for example, the colour must be matched and if it does not match then the point is rejected as not being a line point. If the colour does match, then other feature measures are used for verification. Correct detection of line points with minimum false acceptance is important since these detected points will form the basis of subsequent line tracing.

One example of the line detection routine is as follows:

(1) Scan the image in a raster-scanning manner - from left to right, top to bottom.

(2) For each pixel:

(i) Check against the line centre colour clusters by verifying the probability of that point belongs to the cluster with known mean M and deviation D lies within an acceptable range. This can be implemented by assuming a normal distribution. The probability of a line point X belonging to that cluster can be calculated by the Gaussian distribution function $G_{(M,D)}(X)$.

(ii) If the pixel does not fall into any cluster, go to the next pixel.

(iii) Else, verify the profile the same way as colour.

(iv) If this fails, go to the next pixel.

(v) Else, record the pixel as a line point.

Detection Phase - Line Tracing

The object of line tracing (step 230) is to extract a complete line assuming that there is at least one line point as a starting point for each tracing. The line tracing process will use the prototype information generated in sample collection to verify if the next candidate point is a line point. Line tracing is performed by a four-round algorithm as shown in Fig. 4. In the cases of

automatic/batch mode, the line tracing process starts when the line point detection process is completed. Depending on the memory size of the system and the size of the image being vectorized, it will normally be more efficient to divide the image into a number of blocks and perform line tracing within each block before moving on to the next block.

In the first round (step 310) all previously detected points are linked within an 8-neighbourhood (the eight orthogonally and diagonally adjacent points, see Digital Picture Processing by A. Rosenfeld; Academic Press, 1982). In the second round (step 320) all of the potential line points within a certain look-ahead window are matched against the prototypes and compared with prediction based on known previous line points. The look-ahead window is a rectangle with its long axis aligned with the direction of the line and starting from the last point. A typical look ahead-window size is 5 pixel long and 5 or 3 pixel wide. This process is repeated in the third round (step 330) but with looser conditions. For example, in the second round, the candidates of line points are verified against all chosen conditions such as colour, profile, orientation and width using a suitable similarity function. Since there is no line point identified in the second round, the verification condition is loosened by excluding "profile" in the third round with a hope that a line point may be found using another similarity function. In the fourth round (step 340) two nearby line segments can be merged into one.

Fig. 5 shows an exemplary line tracing process is more detail. Each potential line point in the look ahead window is analyzed as follows. Firstly (step 510) is it the end of another line segment? If it is, and if other criteria are satisfied this can lead to the merging of line segments. If the line point is a single point the similarity of colour and profile between the point under consideration and the preceding line to which it might be added is calculated using the similarity function of Eq. 2 at step 520 and if the similarity is the best so far of all possible line points it is recorded at steps 522, 524 and the loop repeats at steps 530, 540. The same comparison is made when the point under consideration is not the end point of another line segment, the difference being in that case that the current line segment be extended if the point is the best comparison to date at steps 512 - 518, the loop repeats at steps 530, 540. When all points on the look-ahead window have been analyzed in this way, if another segment has been found, this is merged with the existing segment at steps 550, 560. Otherwise, a new point is added to the line segment and the intermediate points between the line segment and the new point are also added at steps 550, 570. The steps shown in Fig. 5 are used for both the second and third rounds of Fig. 4, except that in the third round, the calculations of steps 514, 520 are reduced by a simple calculation of colour similarity, as a looser condition.

Detection Phase: Identification and Correction of Possible Errors

The tracing process as described above is automatic and requires no user input. It will not, however, be sufficient to correctly identify all points and all lines. There will still be some unsolved problems such as line breaks, as well as errors such as line merging and other objects. User interaction at step 240 can correct these errors, solve unsolved problems and so on, and by appropriate design the system can minimise the man-time needed at this stage. An interactive verification and editing module as shown in Fig. 6 can be used here.

This module automatically suggests the possible linkage of any line break at step 610. All line breaks and detected errors are treated as problem points and are prompted to the user one by one. For line breaks, possible links are suggested to the user automatically for verification. The selection of a possible link by the module is based on the distance between two break points and the directions of two line segments at the two break points. When connecting two end points of stringed lines a curve is used at step 620 instead of a straight line to simulate the lost line segment. A curve can match most cases and help a user to connect breaks more quickly. The curve is a 4 order Spline (see, for example Computer Graphics, Principles and Practice, by Dr. James D. Foley; Addison Wesley, Reading 1995) which is

generated by the fitting of the first N points (say 20 points) on both sides of the lines to be joined.

Another possible source of errors is that annotation characters on the image may be traced in error. These are detected by a character recognition device and identified to the user for deletion at step 630.

Other editing and correction functions are handled by editing and correction module 640. Such a module may, for example, be formed by a proprietary product such as Microstation by Bentley, Inc. or similar to allow the user manually to delete all objects of no interest and to edit manually line merging problems, for example.

Due to unpredictable directions of the tracing results and often the complexity of the background, there may be some zigzags along the line after tracing and some points might be outside of the boundary of the line. The distribution of points along the line may not be appropriate either, with some parts of the line being more intense than others. To overcome these problems a line at step 650 is firstly smoothed by converting the line to a Spline by the fitting of points in the line. This curve is then vectorized into points based on a given digitizing tolerance so that the distribution of points in the line is more even. After smoothing there may still be surplus points or points outside of the boundary of the line. These points are removed by

filtering. The distance of a point to the line is used to decide which point should be removed.

The process described above may be regarded as a batch interactive processing system comprising the following steps: sample data collection and feature space optimisation; line point detection for finding line points as seeds for line tracing; adaptive line tracing; and interactive verification and editing. Of these four steps, the first and last are interactive and require user input, though less input than is required for many prior art systems. The second and third stages are automatic.

In an alternative process the line tracing may be performed interactively. In this alternative process sample data and feature space optimization is performed as before, but the tracing step is an interactive one in which the user may identify a trace start point from which a line trace is performed until there are no more acceptable points at which the trace stops. The tracing restarts when the user identifies a further trace start point. This interactive line tracing is then followed as before by verification and editing.